

# Using CT-ANALYST™ to optimize sensor placement

Jay Boris, Keith Obenschain, Gopal Patnaik

Laboratory for Computational Physics and Fluid Dynamics

U.S. Naval Research Laboratory, Code 6400

Washington DC, USA 20375-5320

## ABSTRACT

Networked groups of sensors that detect Chemical, Biological, and Radiological (CBR) threats are being developed to defend cities and military bases. Due to the high cost and maintenance of these sensors, the number of sensors deployed is limited. It is vital for the sensors to be deployed in optimal locations for these sensors to be effectively used to analyze the scope of the threat. A genetic algorithm, along with the instantaneous plume prediction capabilities of CT-Analyst has been developed to meet these goals. CT-Analyst's time dependant plumes, upwind danger zone, and sensor capabilities are used to determine the fitness of sensor networks generated by the genetic algorithm. The optimization and the requirements for the evaluation of sensor networks in an urban region are examined along with the number of sensors required to detect these plumes.

## INTRODUCTION

With the development of networked chemical sensors, and their possible deployment in cities and bases, it is vital to deploy them in optimal locations to provide the most beneficial effect. The characteristics of a sensor network, and the placement of sensors within the network, need to be evaluated for performance for a given situation. A sensor network should be capable of minimizing the detection delay of a source release. This maximizes the response time that people within the area effected have, allowing them to take the appropriate measures to limit their exposure to the release.

The costs and logistics of running, building, and maintaining a sensor network makes it difficult to provide zero detection delay if point detectors are used exclusively. While some delay may be tolerated, it is our goal to minimize this delay within other constraints of the situation. This paper talks about a method using a genetic algorithm to optimize the location and number of sensors required for a given region. CT-Analyst's unique capabilities are exploited in this optimization technique.

## BACKGROUND

### Genetic Algorithms

To find an optimal sensor network, a genetic algorithm using CT-Analyst's features is an attractive potential solution. Since its development by Holland in the 1960's [1], the genetic algorithm has been used successfully in many different fields. Genetic algorithms are a type of search algorithm that works particularly well if the search space is too large to run every potential case and when local maxima exist. For example, to exhaustively search every possible location of a group of 20 sensors in a grid of 350x350 potential locations at a rate of 20 evaluations per second would take months if not years. While the answer generated by a genetic algorithm might not be the best solution, it will typically be a very close approximation to it. The main disadvantage of genetic algorithms is that they potentially require a lot of time and computing resources, depending on the rate of convergence and the computational cost of a fitness function. However, given the amount of time required to evaluate a typical population, many examples of parallelized genetic algorithms exist.

A genetic algorithm evaluates the fitness of genomes in a population, and generates the next population based on the fitness of the previous generation. Each genome is a potential solution to the problem, where the elements of the solution are equivalent to chromosomes in the genome. The initial population is usually chosen randomly, but the initial population can also be seeded with solutions that are known to produce good results. The next population of genomes is determined by combining members of the current population to produce offspring that are based on the scores of each parent genome's fitness function. This is known as crossover. During crossover, individual chromosomes within the offspring can potentially mutate, giving the offspring slightly different characteristics that are unique from its parents. This is particularly useful in later generations of the population, where the population is fairly homogeneous. The user

determines the fitness function of a genome, in which the performance of a genome is evaluated, and a fitness score is assigned. Members with a high fitness score will typically have many offspring in the next generation while those with a low fitness score could have few or none. New populations are generated, and evaluated until one of several requirements is met. This includes the desired fitness level of a member of the population, the average fitness of the population has reached some level, or the maximum number of generations has been calculated.

An approach using genetic algorithms was selected for sensor optimization because the characteristics making up a robust sensor network were largely unknown. This approach also made it easy to modify the characteristics we were looking for while leaving the search method intact. We also believe that the results from this approach can be used to generate a more efficient approach to this problem. Furthermore, advances in contaminant transport modeling made it possible for this search technique to be utilized.

### **Contaminant Transport Modeling**

FAST3D-CT, a detailed physics-based numerical model [2], was developed at the Naval Research Laboratory to accurately predict plume evolution and the contamination footprints resulting from these releases. FAST3D-CT is a general-purpose fully 3D computational fluid dynamics (CFD) model for contaminant transport in complex urban geometries. It is based on the high-resolution, time accurate Flux-Corrected Transport algorithms invented at NRL. A wide range of contaminant physics and physical environment models are employed in FAST3D-CT. FAST3D-CT is used to perform the time-accurate, high-resolution 3D CFD as an offline, high-fidelity urban contaminant transport scenario generator.

The salient data and statistical features from the FAST3D-CT CFD calculations are summarized and distilled into memory efficient, time-independent Dispersion Nomograph<sup>TM</sup> data sets. These are interpreted and evaluated by CT-Analyst. There are no time-dependent integrations performed explicitly by CT-Analyst. Instead, all predictions produced by CT-Analyst are simply the result of applying an interpolation procedure utilizing the appropriate nomograph data set based upon the high-resolution CFD results. This same interpolation procedure can be used in both upwind and downwind directions with equal effectiveness. These simple geometric operations are used to determine the probable source zone upwind of each sensor. CT-Analyst provides the unique capability to immediately backtrack and simultaneously determine the location of multiple unknown sources simply based on sensor readings and meteorological parameters.

CT-Analyst has undergone several years of stringent testing and meets the necessary real-world and real-time requirements for an urban airborne contaminant emergency assessment tool. Validation studies of the components of this new technology and field trials on “beta test” have been successfully conducted. The plume “predictions” from CT-Analyst, based on a quantitative Figure of Merit, agree, within 80 to 90%, with the FAST3D-CT CFD simulations on which they are based and yet are available much faster than the corresponding Gaussian plume estimates [3]. Furthermore, the underlying CFD technology is uniformly convergent so answers automatically get better with increasing computer power because higher resolution CFD simulations can be used to build the Dispersion Nomograph data sets [4]. In addition, the resulting capability includes necessary, crucial features not found elsewhere [5].

The existing plume prediction technology in use throughout the nation is based on Gaussian similarity solutions (“puffs”), an extended Lagrangian approximation that only really applies for large regions and flat terrain where large-scale vortex shedding from buildings, cliffs, or mountains is absent. These models, used in a number of implementations in the U.S. and throughout the world, fall prey to an important dilemma that the CT-Analyst approach avoids. In these Gaussian models, improving the quality of the answer requires considerably more computer time per run. Reducing the run time requires reducing the quality (accuracy) of the answer.

The use of CFD models or Gaussian plume models are not suitable for use as the fitness evaluation of a genetic algorithm due to their relatively long times to generate plumes, and the sheer number (many millions) of fitness evaluations and iterations required for a solution to converge. Even if the time to generate a Gaussian plume decreased significantly, the plumes generated would not take into account the 3D geometry of an urban region. CT-Analyst's plume capability is well-suited for this type of evaluation because it produces plumes comparable to the CFD calculation as stated above while producing this result in about one millionth of the time [5]. The speed of CT-Analyst allows fitness functions to be evaluated for performance quickly. Table 1 shows the approximate amount of time required to run a genetic algorithm for 1000 generations using various plume models.

<i>Plume model</i>	<i>Computer</i>	<i>Approximate time to run a fitness evaluation for 1000 generations (population = 1000)</i>
CFD(FAST3D-CT)	Supercomputer	~9000 hours (random sources)
Gaussian	Workstation	~500 hours (random sources)
CT-Analyst	Laptop	~33 hours (random sources)
CT-Analyst	Laptop	~4 hours (time dependent sensor coverage, 20 sensors)

Table 1. The approximate time and the generic computer required using a genetic algorithm for this problem.

## APPLICATION OF GENETIC ALGORITHMS FOR OPTIMAL SENSOR PLACEMENT

An elite simple genetic algorithm was used [1] where the members of the population with the highest fitness scores were kept in the next population. This ensures that the population's maximum fitness score will not decrease and also reduces the number of generations required to converge to an answer [1]. The rate of crossover was set at 0.95 with the rate of mutation set at 0.25 percent, where the mutation increased if the rate of convergence decreased by a threshold. In our case the genome was the set of locations of the sensors in the sensor network with the chromosomes consisting of (x, y) coordinates of the sensors. The population size was set to 1000. While the individual fitness function is now relatively fast, the algorithm was distributed over multiple processors using MPI. The evaluations of the population are spread out over multiple processors, with the best results of a generation saved as candidates for the solution. The machine used for this implementation is a cluster of dual-processor Opteron. This algorithm is compute bound so a high-speed interconnect is not necessary. Several different approaches were examined for the fitness function.

The first approach uses a plume model to generate plumes from randomly placed sources and then analyzes the sensor network's ability to detect the plume within time  $t$  of release. In this case, if a least one sensor is located within the plume, it counts as a detection of the plume. The sensor network evaluates individually a sequence of randomly located sources, with the fitness score based on the total number of sources detected. A new set of random sources must be calculated for each generation. If the set of source locations is fixed, the sensor network's solution would converge on the coverage of that set of fixed sources, but not on a optimal coverage of sources located anywhere in the region. This method has the advantage of being able to use a variety of plume prediction tools like Gaussian plume models, CFD models (e.g. FAST3D-CT), and Dispersion Nomograph tools (e.g. CT-Analyst). However CT-Analyst is the best choice due to its speed and accuracy (Table 1, lines 1-3).

While this approach is acceptable, a much more efficient procedure was developed using CT-Analyst unique upwind capability. Figure 1a (next page) shows an example of the upwind danger zone for site. The corresponding upwind zone for a sensor represents the upwind area where the contaminant from a source could hit the sensor. This upwind, probable source zone or "backtrack" zone was modified to be time-dependent and can also be described as an "anti-plume". Sensor coverage is the union of the "anti-plumes" for all of the sensors in the region. Figure 1b illustrates this updated display. Using the union of anti-plumes as the fitness function decreases the time to evaluate a sensor network for a region drastically (see Table 1, line 4). The new fitness function is now the total area of sensor coverage for a given region ranging from zero to one, which could be calculated with a single call to CT-Analyst.

Because of the increase in efficiency, the second approach was selected for the main optimization trials. To determine the optimal amount of sensors required for this region, sensor networks from five to forty sensors, in five sensor number increments were evaluated for total sensor coverage on a 2 km by 2 km region for a typical city. The wind was from the northwest, with a speed of three meters per second. The region itself is an urban area with varying degrees of building density ranging from open areas free of structures to city blocks with high building density. The Dispersion Nomograph utilized for this region was generated using FAST3D-CT, which includes all of the effects of buildings, streets, trees, etc. A version of CT-Analyst was developed to execute without the GUI using a prototype of the CT-Analyst Application Programmer's Interface (API). CT-Analyst is used to evaluate sensor configurations for a detection delay of three minutes, six minutes, and nine minutes. These times were selected based on results obtained from the walk away program. In reference [5] nine minutes warning delay is found to be maximum delay to be tolerated if at least 50% of a population in an area affected by a moderately large plume is to be saved.

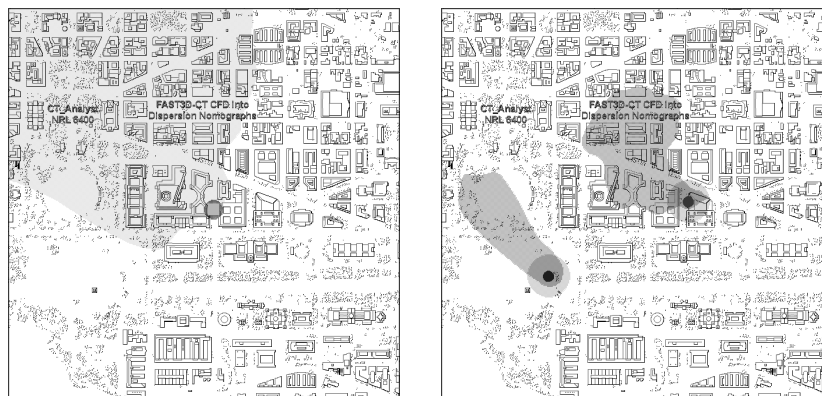


Figure 1. 1a. Upwind danger zone from site - infinite time. 1b. Upwind, time dependent “anti-plume” from sensor.

## RESULTS

Figure 2 below shows the fractional area covered versus number of sensors for detection delay of three, six, and nine minutes. The number of sensors required producing adequate coverage increases significantly as the plumes size decreases. Only 10 to 15 sensors are required to obtain 90% coverage for a nine minute time delay, contrasted with over 40 for a three minute delay. Even with 50 sensors, complete coverage of the region can not be obtained for the three-minute delay while additional sensors became completely redundant past 30 sensors for these six- and nine-minute warnings.

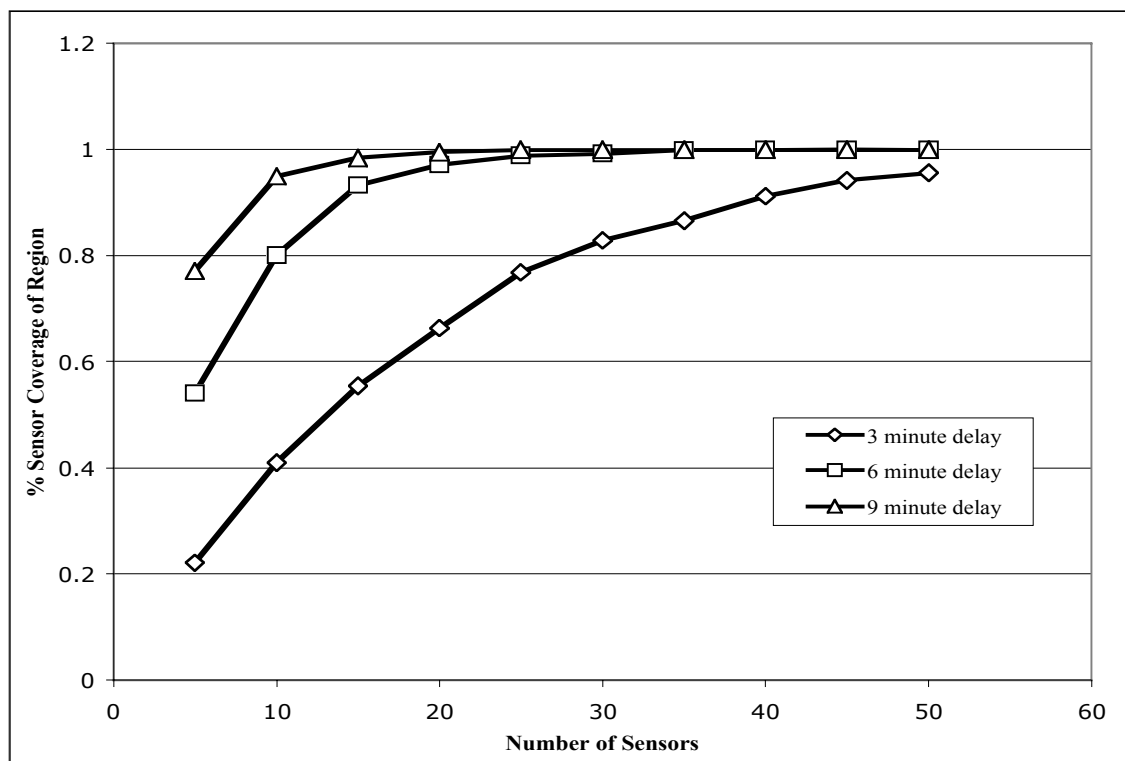


Figure 2. Fractional area of sensor coverage as a function of the number of sensors ( 2 km x 2 km coverage area).

Figures 3a and 3b represent the minimal sensor network required to detect at least 90% of the region for three- and nine-minute detection delays. For a nine-minute delay (Figure 3b), sensors are placed towards the edge of the region, opposite of the wind direction because at nine minutes the “anti-plumes” are very large, and sensors are wasted if they are placed further upwind. If the time delay for detecting a plume is increased beyond nine minutes, the eventual result is a sensor network with all of the sensors placed along the edge of the domain. 40 sensors are required to provide the same coverage if a three-minute detection delay (figure 3a) must be provided. The density of sensors for a given area in the region varied. More sensors were required for relatively open areas and where the plume funneled through gaps between buildings. This was particularly noticeable when the time delay allowed for detecting plumes was short.

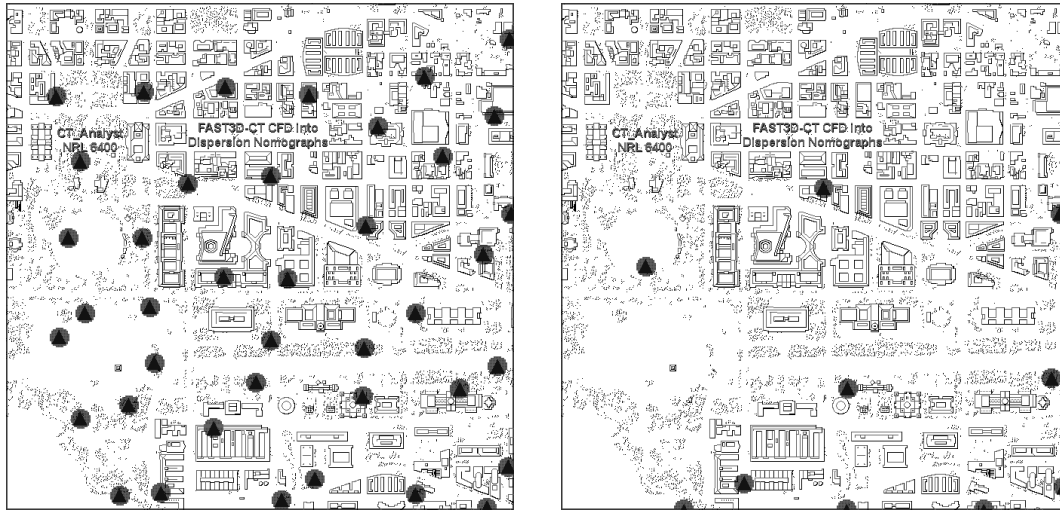
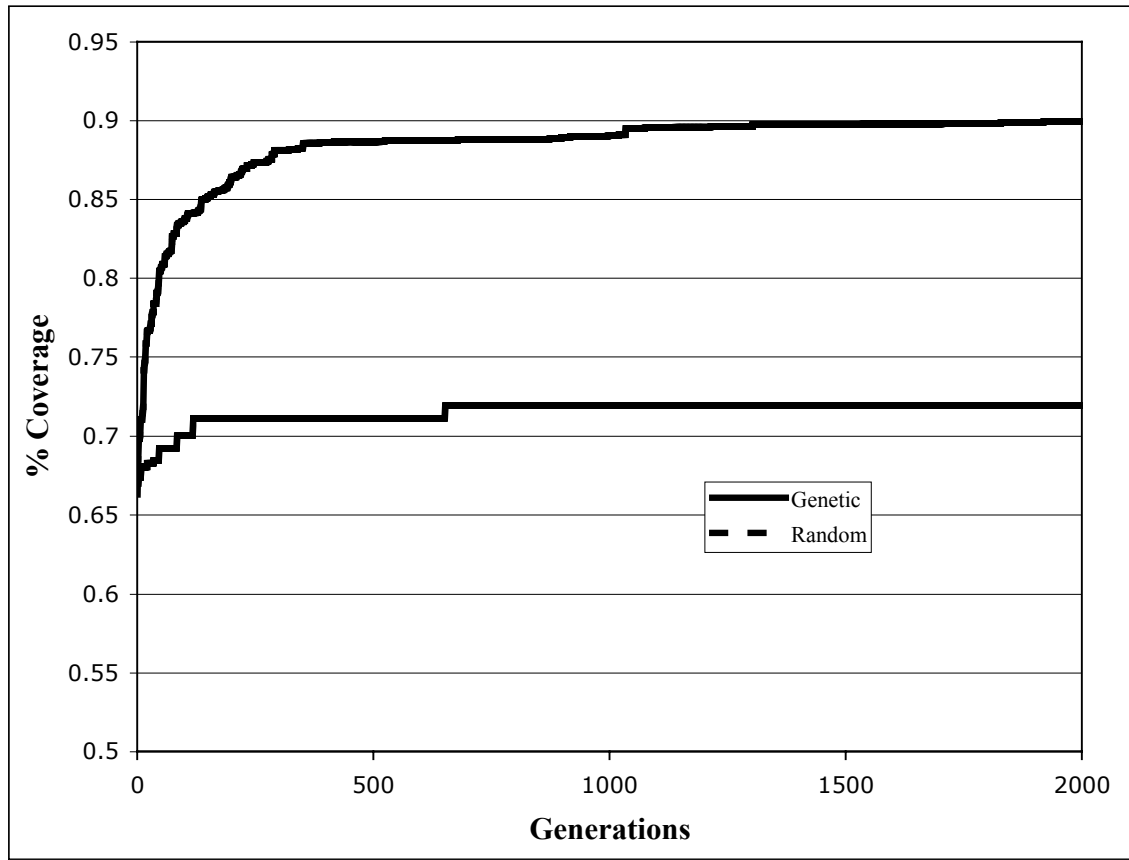


Figure 3a, and 3b. The solutions for a three-minute and nine-minute detection delay respectively.

The shape of the plume envelope can explain this result. In areas with few buildings, the plume envelopes are narrow and elongated, looking very much like their Gaussian plume counterparts. In areas with many buildings, the shape of the plume envelope is broader, depending on the geometry of the buildings and wind angle. Figures 4a and 4b depict plume envelopes for the release of two sources in the region after three and after nine minutes. The first source is released in an open region while the second is in an area with high building density. Note that the narrow plume's shape starts to change (figure 4b) as it encounters a city block with high building density. In order to detect a narrow plume more sensors are required.



Figures 4a (three-minute plume), and Figure 4b (nine-minute plume)



Figures 5. Fractional coverage comparing genetic algorithm and random source approaches.

Figure 5 shows the coverage of the sensor network versus a random sensor placement run for the same number of intervals. The random (brute force) sensor placement is evaluated in the same manner as the genetic algorithm with the best candidate produced of each generation reported as the maximum coverage attained. For the same amount of effort, here two million calls to CT-Analyst, the generic algorithm covered over 90% of the region while the random-placement approach's best answer results in coverage of about 72% of the region.

### Conclusion

The use of a genetic algorithm to produce a plausible and useful sensor optimization has been shown. This approach was not possible until CT-Analyst's low-latency evaluation of contaminated regions was developed. To calculate 1000 generations requires 1 million calls to CT-Analyst and many millions of individual sensor backtrack "anti-plume" evaluations. With more complex fitness functions, and more stringent requirements for a sensor network, the time to calculate an optimal network will only increase. Use of other plume models is prohibitive. This approach is one technique for determining the optimal sensor placement. It has also shown that to provide guaranteed short detection delays will require many sensors. This calculation assumes that of the sensors are always accurate. To make a robust network that can handle false alarms and errant reports will also increase the total amount of sensors required.

### Future Work

While this method easily finds a functional sensor network for one wind angle, it needs to be extended for multiple wind angles. Using regional information on seasonal average wind direction, and speed can be used to weight the importance of wind in the placement of sensors. Sensor placement also needs to be verified to see how differing criteria can change the optimal location of the sensors. For instance, if CT-Analyst's ability to backtrack a source is used in the fitness function, with the accuracy of backtrack weighted heavily for a sensor network's score, how different is the result

compared to the sensor network's coverage based solely on the wind angle and speed? These two results will be presented at the conference. Other variables could also be added to the genome. As well as the sensor location, the type of sensor deployed could be added to the genome of the members in a population. This should also allow including considerations of redundancy for treatment of false alarm possibilities.

The variant of the genetic algorithm used here has a simple implementation. Other mechanisms used with genetic algorithms need to be evaluated to determine how they affect the convergence of a solution as well as the final solution's performance. This includes changes in how the algorithm is parallelized, the selection of mates for crossover, and varying techniques of modifying when crossover and mutation occur.

This new method also treats the nomographs and the individual sensor placement information as externally given functions. The results obtained from this approach will have to be examined to determine how the properties of the Dispersion Nomographs themselves, as well as other information in a region, can affect rules of thumb for the placement of sensors. The results from the various runs of this genetic algorithm will need to be analyzed further from this perspective. Even if other direct algorithms do not provide results that outperform the genetic algorithm, they could still be used to seed the initial population for the genetic algorithm. This could further speed up the convergence.

## References

1. Goldberg, David E. **GENETIC ALGORITHMS in Search, Optimization and Machine Learning**, Addison Wesley Longman, Incorporated, 1989.
2. J.P. Boris, "The Threat of Chemical and Biological Terrorism: Preparing a Response," **Computing in Science and Engineering** **4(2)**, pp. 22 – 32, March/April 2002.
3. J.P. Boris, K. Obenschain, G. Patnaik, and T.R. Young (2002) "CT-ANALYST™, Fast And Accurate CBR Emergency Assessment", **Proceedings of the 2<sup>nd</sup> International Conference on Battle Management**, Williamsburg, VA, 4-8 November 2002.
4. G. Patnaik, J.P. Boris, F.F. Grinstein, and J.P. Iselin Large Scale Urban Simulations with the MILES Approach." **AIAA Paper 2003--4104**, AIAA CFD Conference, Orlando FL, 25 June 2003.
5. J.P. Boris, J. Fulton, K. Obenschain, G. Patnaik, and T.R. Young, "CT-ANALYST, Fast and Accurate CBR Emergency Assessment", **Proceedings: SPIE Defense and Simulation Symposium**, SPIE Paper 5416-01, Orlando FL, 10-16 April 2004.